Draft

Encompassing Colored Crossing-Free Geometric Graphs*

Ferran Hurtado[†] Mikio Kano[‡] David Rappaport[§] Csaba D. Tóth[¶]

Abstract

Given n red and n blue points in the plane and a planar straight line matching between the red and the blue points, the matching can be extended into a bipartite planar straight line spanning tree. That is, any red-blue planar matching can be completed into a crossing-free red-blue spanning tree. Such a tree can be constructed in $O(n \log n)$ time.

keywords: geometric graph, spanning tree, color

1 Introduction

Interconnection graphs among disjoint objects in the plane are fundamental in computational geometry, the geometric TSP being a flagship example. Since a minimum length TSP tour of points in the plane has no self-crossing, interconnection graphs are often thought of as *planar straight line graphs (PSLGs).* Numerous variants of interconnection graph problems were studied in recent years, including Hamiltonian tours, Hamiltonian paths, and spanning trees satisfying various constraints.

This paper addresses two problems on connecting disjoint components of a planar straight line graph. The first problem involves color conforming augmentation of colored graphs into connected PSLGs.

A second problem is concerned with the augmentation of 2-edge connected (but monochromatic) PSLGs. A connected graph is *2-edge connected* if at least two edges need to be removed to split the graph into two or more connected components. We have the following results.

• Consider a PSLG G and suppose it has k connected components, and no component is a single vertex. Furthermore, the vertices of G are colored so that no two adjacent have the same color. See Fig. 1. We show that one can add $k-1$ straight line edges to G so that we obtain a connected PSLG that conforms to the coloring.

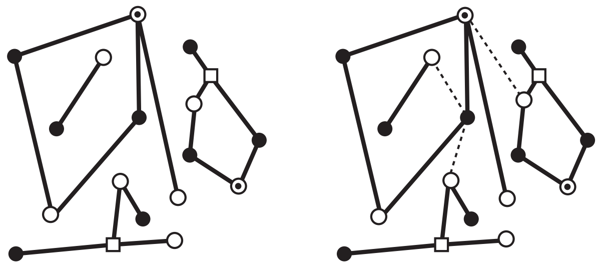

Figure 1: Augmenting a colored disconnected PSLG.

• In particular, if we are given a set of n bi-chromatic line segments, we can find a set of $n-1$ edges so that we are left with a color conforming planar straight line spanning tree. See Fig. 2.

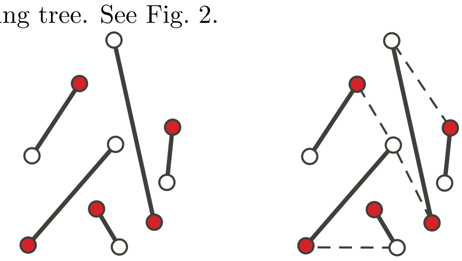

Figure 2: Augmenting disjoint bi-chromatic segments.

• Suppose G is a PSLG consisting of k 2-edge connected components. We can add $2(k-1)$ edges to G so that the result is a 2-edge connected PSLG.

• In particular, we can augment a set of k disjoint triangles with $2(k-1)$ line segments leaving a 2-edge connected PSLG such that every bounded face is a triangle. See Fig. 3.

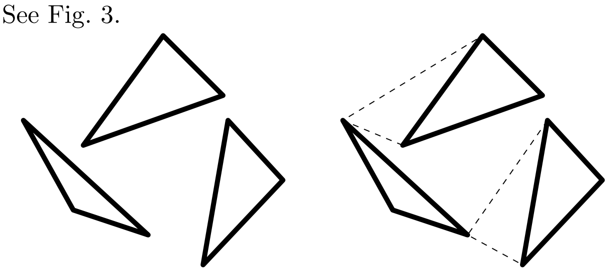

Figure 3: Augmenting a set of triangles to obtain a 2-edge connected PSLG such that every bounded face is a triangle.

We offer a constructive proof for all the above problems based on the following theorem. In what follows

*Full version. An extended abstract appears in *Proc. CCCG.*

†Dept. Matèmatica Aplicada II, Universitat Politècnica de Catalunya, Barcelona, Ferran.Hurtado@upc.es supported by projects DURSI 2001SGR00224 and MCYT BFM2003-0368.

‡Department of Computer and Information Sciences, Ibaraki University, Hitachi, kano@cis.ibaraki.ac.jp.

§School of Computing, Queen's University, Kingston, ON, daver@cs.queensu.ca supported by NSERC of Canada Discovery Grant 9204.

¶Department of Computer Science, University of California at Santa Barbara, Santa Barbara, toth@cs.ucsb.edu supported by NSF grants CCR-9901958 and IIS-0121562.

we make the simplifying assumption that no PSLG contains a component consisting of a single vertex.

Theorem 1 *For any two finite PSLGs, whose planar drawings are disjoint, one of the graphs has a vertex that sees an entire edge of the other graph.*

Note that the roles of two PSLGs, A and B, in Theorem 1 are not symmetric: It is possible that A has no vertex that would see an entire edge of B; in this case a vertex of B sees a full edge of A. In Section 2, we show that if the convex hull of (drawing of) B does not contain A then a vertex of A sees an entire edge of B.

Theorem 1 leads to an algorithm to construct a color conforming spanning tree in the first problem and a 2-edge connected augmented graph in the second problem. In Section 4, we provide an alternative proof for the first problem that can be turned into a $O(n \log n)$ time algorithm.

Theorem 2 *Every set of bi-chromatic line segments, where any two segments are either disjoint or share an endpoint, can be extended to a color conforming and connected PSLG in $O(n \log n)$ time.*

1.1 Related previous results

Colored PSLGs. Geometric graphs on red-blue points have received increasing attention recently. For a set R of red and B of blue points in the plane, $K(R, B)$ denotes the geometric bipartite graph whose vertex set is $R \cup B$ and whose edges are the red-blue line segments. A path in $K(R, B)$ is necessarily *alternating* between red and blue points. It is well known that for n red and n blue points in the plane, there is always a crossing free perfect red-blue matching (e.g., by repeated application of the *ham sandwich* theorem [11]).

For n red and n blue points in the plane, $K(R, B)$ does not always contain a crossing-free Hamiltonian tour [1]. Kaneko, Kano, and Yoshimoto [10] proved that such a Hamiltonian tour have $n - 1$ self-crossings in the worst case. Kaneko and Kano [9] showed that if $|R| = \Theta(|B|^2)$ then there is an alternating path containing all *red* points. Kaneko [7] proved that for any n red and n blue points in the plane, there is a color conforming connected PLSG of maximal degree three.

These and many other interesting results on geometric red-blue graphs can be found in a recent survey paper of Kaneko and Kano [8].

Encompassing graphs. Given a set of pairwise disjoint line segments in the plane, an *encompassing tree* is a PSLG whose vertex set is the set of segment endpoints and contains every input segment as an edge.

Notice that every encompassing path consists of input segments and non-input segments alternately. Not every set of segments admits a Hamiltonian encompassing path. Pach and Rivera-Campo [12] showed that every set of n segments have a subset of size $\Omega(n^{1/3})$ for which an Hamiltonian encompassing tree exists. The longest alternating path not crossing any of the initial n segments has size $\Theta(\log n)$ in the worst case [5]. Bose, Houle, and Toussaint [3] proved that every set of disjoint line segments in the plane can be augmented to a connected PSLG of maximal degree three. They can construct such a tree for n segments in $O(n \log n)$ time. Later, Hoffmann and Tóth [6] proved that there is also an Hamiltonian encompassing graph of maximal degree three.

2 Proof of Theorem 1

In order to prove Theorem 1 make use of the following algorithm.

Input: A connected PSLG A, and a PSLG B such that A is not completely contained in the convex hull of B, that is, there a point of A incident to the convex hull of $A \cup B$.

Output:A vertex α in A that sees the segment $\beta\gamma$ from B.

Initialize It is well known that a PSLG, may be augmented with a set of line segments to obtain a triangulation of the convex hull of the drawing See [2]. This implies that there is a vertex of A that sees a vertex in B. Let α be a vertex in A that sees a vertex from B, β. Let γ denote a vertex in B such that $\beta\gamma$ is an edge in B, and the interior of the triangle $\alpha\beta\gamma$ does not contain a point of any line segment of B incident to β, as shown in Fig. 4. As a notational convenience let $\Gamma(\alpha, \beta)$ denote a function that returns such a point γ.

Iterate Sweep r a ray with origin α passing through β from β to γ until it hits δ the first vertex of A or B visible to α. We maintain the invariant that α is a point in A that sees β a point in B.

case 1. $\delta = \gamma$ We are done, α sees the segment $\beta\gamma$. Output α sees $\beta\gamma$.

case 2. $\delta \neq \gamma$ and δ is a vertex of B] Set $\beta \leftarrow \delta$ and $\gamma \leftarrow \Gamma(\alpha, \beta)$. Iterate.

case 3. $\delta \neq \gamma$ and δ is a vertex of A] Set $\alpha \leftarrow \delta$ and $\gamma \leftarrow \Gamma(\alpha, \beta)$. Iterate.

We use the notation $\alpha_i, \beta_i, \gamma_i, \delta_i$ to denote the corresponding values at iteration i. Without loss of generality we assume that r sweeps in an anti-clockwise direction. We make the following observations.

Observation 1. If $\alpha_i \neq \alpha_{i+1}$ the triangle $\alpha_i\beta_i\alpha_{i+1}$ is empty, that is, the interior of this triangle avoids both A and B. Furthermore, the line through $\alpha_i\alpha_{i+1}$ intersects the line segment $\beta_i\gamma_i$ at a point δ'_i such that the triangle $\alpha_i\beta_i\delta'_i$ is empty. We refer to this triangle as Ti. See Fig. 4

Observation 2. If $\beta_i \neq \beta_{i+1}$ the triangle $\beta_i\beta_{i+1}\alpha_i$ is empty. Furthermore, the line through $\alpha_i\beta_{i+1}$ intersects the line segment $\beta_i\gamma_i$ at a point δ'_i such that the triangle

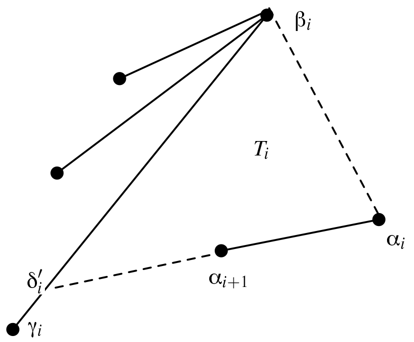

Figure 4: The value of $\gamma = \Gamma(\alpha, \beta)$ is illustrated. The empty triangle $T_i = \alpha_i\beta_i\delta'_i$ is also shown.

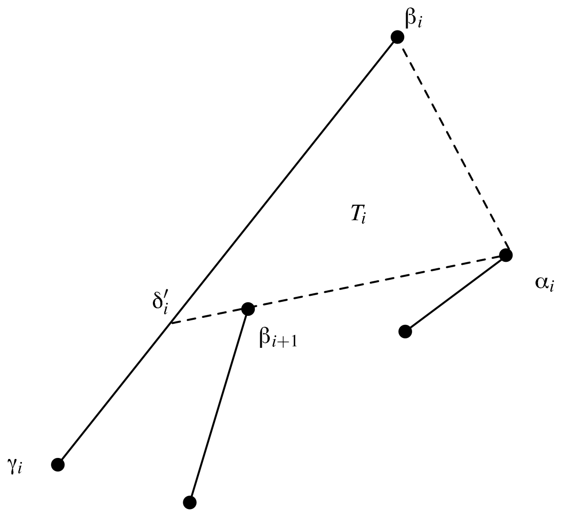

Figure 5: Empty triangle $T_i = \alpha_i\beta_i\delta'_i$

$\alpha_i\beta_i\delta'_{i+1}$ is empty. We refer to this triangle as Ti. See Fig. 5

From observation 1. we see that the iteration invariant is maintained, that is, at the start of every iteration α is a point in A that sees β a point in B. Thus if the algorithm terminate it returns the correct result.

It remains to show that this iterative process terminates. We argue that non-termination contradicts our initial assumptions that A is connected and has a point on the convex hull of $A \cup B$.

Consider the possibility that the algorithm does not terminate. That is, we enter a cycle of vertices of A and B that are repeatedly visited. Let us examine a minimal cycle where at some iteration j we have a vertex α_j in A, that sees β_j in B, and at some iteration k we enter a state where $\alpha_k = \alpha_j$ and $\beta_k = \beta_j$. Thus the sweep of r makes a full circle. Since r sweeps through a full circle the δ values must lie on at least three distinct segments from B, so we visit at least three distinct vertices from B. By observations 1 and 2 above we see that there is an empty triangle T_i for $i, j \le i \le k$. Consider the region bounded by the union of these triangles, and call it Q. See Fig 6. From observation 1, that is, the fact that extensions through $\alpha_i\alpha_{i+1}$ intersect $\beta\gamma$ we infer that outer boundary of Q contains the vertices from A that take on α values. From observation 2, that is, the fact that the segment $\beta_{i+1}\delta'_i$ does not cross a segment from A we infer that the outer boundary separates the α vertices from any vertex of A that is on the convex hull of $A \cup B$. This establishes the desired contradiction.

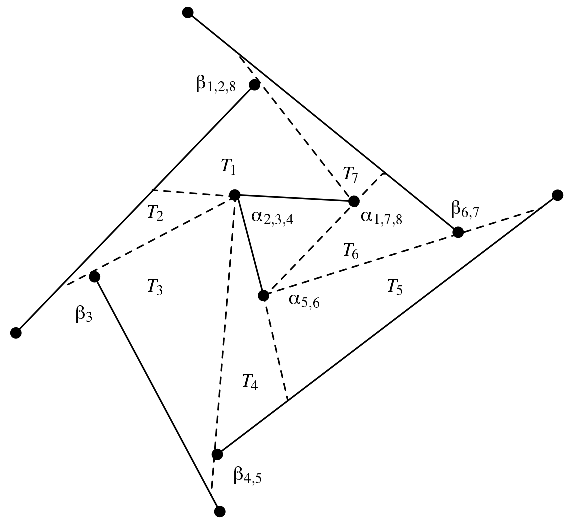

Figure 6: The list of numbers associated with each vertex represents the iteration(s) where that vertex was current. Non termination of the algorithm contradicts our assumption that A is connected and not contained in the convex hull of B.

Thus we conclude that the algorithm does indeed terminate and that there is a vertex in A that sees an entire segment from B. This fact is summarized by the following lemma.

Lemma 3 *Let A be a connected PSLG , and B a PSLG such that A is not completely contained in the convex hull of B, that is, there a point of A incident to the convex hull of $A \cup B$. Then there is a vertex of A that sees an entire segment of B.*

We now establish Theorem 1.

Proof of Theorem 1. Consider two PSLGs A and B whose planar drawings are disjoint. Assume that a vertex of the convex hull of $A \cup B$ is a vertex of A (in other words, the convex hull of B does not contain that of A). We show that a vertex of A sees an entire edge of B.

Lemma 3 brings us most of the way to proving the theorem. We only need to consider the cases where A is not connected.

Let A' be a connected component of A incident to the convex hull of $A \cup B$, and let $B' = B \cup A \setminus A'$. We proceed by induction on the cardinality $|A \setminus A'|$. We know that a vertex a' of A' sees an entire edge e of B'. Our proof is complete if e is an edge of B. Otherwise e is an edge between two components of A. By induction, the theorem holds for $A \cup \{e\}$ and B, and so we conclude that a vertex of A sees an edge of B.

□

3 Applications

For our first application we consider a plane drawing of a graph with k connected components and with vertices colored so that no edge of the graph is monochromatic. We want to add $k-1$ edges so that we are left with one single connected component with no monochromatic edges.

We proceed by induction on the number of components. If there is only one component, then the input graph is connected. Otherwise we partition the input in two disjoint parts which we call A and B. It follows from Theorem 1 that a vertex v of A or B sees at least one entire edge of B or A, respectively. Since no edge is monochromatic, either $\{a, w\}$, or $\{a, u\}$ is a color conforming connection between A and B. Augment the input graph by this edge: the number of connected components drops by one—induction completes the proof.

For our second result, assume that we have a planar drawing of a graph such that each component of the graph is 2-edge connected. One example of such an input is a set of disjoint triangles. Suppose that there are k connected components in the input. We want to augment this drawing with $2(k-1)$ edges so that we have a single component that is 2-edge connected. This problem closely resembles the problem above where we augment a colored straight line drawing of a planar graph. The only difference is that we connect a vertex to both endpoints of the visible line segment.

4 Proof of Theorem 2

Our second proof for the first problem is similar in some sense to that of Bose et al. [3]: We construct a convex partition of the free space around the line segments, and then add non-crossing edges in each of the convex faces. Since the number of vertices lying along a single face can be arbitrary and we have no control over the distribution of red and blue segment endpoints incident to a single cell, we cannot give a bound on the maximal degree of the resulting spanning graph.

4.1 Convex partitioning

Assume that we are given n pairwise disjoint segments in the plane. We assume, for simplicity, that no segment is vertical and there are no two collinear segments. The free space around the segments can be partitioned into $n+1$ convex cells by the following two-phase partitioning algorithm (Bose et al. [3] used a similar partition):

In the first phase, we sweep the plane from left to right. We extend every input segment beyond its *right* endpoint simultaneously to the sweep line. If an extension hits an input segment, then it stops there. If two extensions meet then they are merged into one extension as follows: if their slopes have opposite signs then both extension continue as a horizontal extension; if they have the same sign then the extension whose slope has smaller absolute value continues and the other stops. In the second phase, every segment is extended beyond its *left* endpoint in a right-to-left plane sweep. An extension stops if it hits an input segment or a previous extension. We apply the same rules as in the first phase if two extensions meet. (See Fig. 7 for an example.)

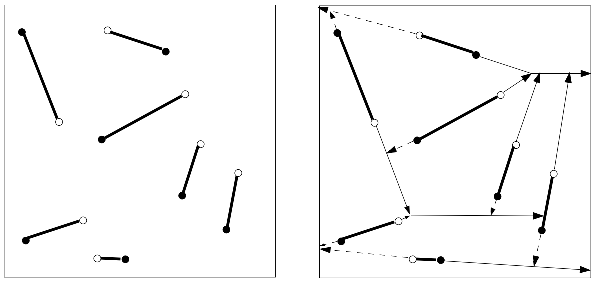

Figure 7: Disjoint segments and a convex partition.

The segments and their extensions form a cell complex in the plane. Every cell is convex and the number of cells is exactly $n+1$. We say that a portion of an input segment (or an extension) is an *edge of the complex* if it lies on the common boundary of two cells. The *vertices of the complex* are the segment endpoints and points lying on the common boundary of three cells. Observe that none of the edges of the cell complex is vertical.

We define an orientation on the edges: The input segments have *no* orientation; an extension edge is oriented left-to-right (resp., right-to-left) if it was created in the left-to-right (resp., right-to-left) plane sweep of the partition algorithm. The in-degree (out-degree) of a vertex of the cell complex is the number of incident edges oriented to (from) it. We use a simple but key property of the cell complex in our main argument:

Lemma 4 *If the boundary of a cell C contains edges of both clockwise and counter-clockwise orientation w.r.t. C, then the boundary of C must contain an entire input segment.*

Proof. Consider two oriented edges on the boundary of C such that their starting vertices are at minimum distance, and let q_1 and q_2 be their starting vertices. By definition of the orientation, the out-degree of every vertex is at most one, and so $q_1 \neq q_2$. Since non-oriented edges are disjoint, q_1 and q_2 are adjacent. The vertices q_1 and q_2 have out-degree one and are incident to the non-oriented edge q_1q_2, therefore they are segment endpoint and q_1q_2 is an input segment on the boundary of C. □

Using the terminology of [3], we call every connected component of oriented edges an *extension tree*. From every point of an extension tree, the orientations lead to a common point (*root*), lying on an input segment or at infinity.

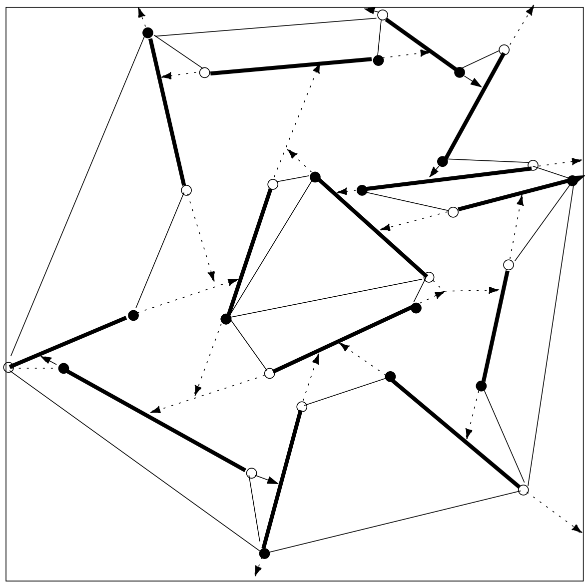

Figure 8: The resulting graph G_1 is not necessarily connected.

4.2 Two phase algorithm

We construct the required graph in two phases.

First phase. Consider a set of bi-chromatic segments and convex partition obtained by the above algorithm. We construct a PSLG G_1 by augmenting the input matching with edges between segment endpoints incident to a common cell. If a cell C is incident to red (resp., blue) vertices only, then we add no edges in C. If a cell C is incident to both red and blue segment endpoints then we connect them by a spanning tree within C:

Lemma 5 *For a set P of a red points and b blue points in convex position, $a, b \geq 1$, one can construct a red-blue planar straight line spanning tree in $O(a+b)$ time.*

Proof. We proceed by induction. If $a = b = 1$ then the edge connecting these two points is the required graph. If $a + b > 2$ then let p and q be two consecutive points with different colors. Assume without loss of generality that there is another p is not the only point of its color. Connect p and q by an edge, and call this algorithm for the set $P \setminus \{p\}$. □

The resulting bipartite PSLG G_1 is not necessarily connected yet. See Fig. 10 for an example. We can establish connectivity of the points lying on a common *extension trees*, though.

Lemma 6 *Every segment endpoint incident to the same extension tree of the cell complex belong to the same connected component of the graph G_1.*

Proof. Let p and q be two segment endpoint incident to the same extension tree T. Let w be the first ancestor of p and q in T (that is the first common vertex of the directed path from p and q to the root of T). Finally let $p = p_1, p_2, \ldots, p_k = q$ be the descendant leaf vertices of w ordered in a BFS traversal of T. (See Fig. 9. left.)

Notice that p_i and p_{i+1}, $i = 1, 2, \ldots, k-1$. are segment endpoints lying on the boundary of a common convex cell C_i. Furthermore, the boundary of C_i contains edges of opposite orientation w.r.t. C_i, and by Lemma 4 it contains an entire input segment. This implies that every C_i, $i = 1, 2, \ldots, k-1$ is incident to a red and a blue segment endpoint, therefore p_i and p_{i+1} are connected in G_1 along a subgraph lying in C_i. □

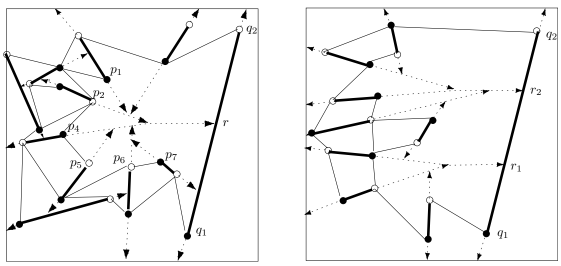

Figure 9: Segments incident to the same extension tree or to extension trees whose root lies on the same segment are connected in G_1.

Similarly, vertices of extension trees whose roots are on the same input segment belong to the same component of G_1.

Lemma 7 *All segment endpoint whose extension tree hits an input segment q_1q_2 on the the same side are in the same connected component of G_1.*

Proof. Let $r_1, r_2, \ldots, r_k \in q_1q_2$ be the roots of distinct extension trees on the left side of q_1q_2. (See Fig. 9, right.) It is sufficient to prove that vertices on two consecutive extension trees rooted at r_j and r_{j+1} are in the same component of G_1. Let C_i denote the cell adjacent to the left side of the edge $r_{i-1}r_i$ (n.b., $r_{i-1}r_i \subset q_1q_2$). This cell contains edges of opposite orientation (incident to r_i and r_{i+1}) and so by Lemma 4 all segment endpoint incident to C_i are connected in the graph G_1. This cell is also incident to segment endpoints whose roots are r_{i-1} and r_i. By Lemma 6, all vertices on the extension trees of r_i and r_{i+1} are connected in G_1 □

Second phase. We add one more edge to the graph for all but one connected component. Let us denote the connected components of G_1 by $L_1, L_2, \ldots, L_h$, ordered according to the x-coordinate of the right-most segment endpoint of each component (i.e., L_1 contains the overall right-most segment endpoint). We describe how to connect the components L_1 and L_2 by one red-blue straight line edge while maintaining a PSLG. Iterating this step leads to the required graph G_2.

Let p denote the right-most segment endpoint of L_2. Consider the extension tree T of p. The root of the extension tree cannot be at infinity, because then T would be incident to a segment endpoint s which lies to the right of p, that is, $s \in L_1$, and by Lemma 6, L_1 and L_2 were connected. Assume that the root of T is r and r lies on a segment q_1q_2. Since $q_1q_2 \in L_1$, the endpoints q_1 and q_2 are not connected to L_2 in the graph G_2.

Let C_1 and C_2 be the cells incident to q_1 and the q_2, resp., on the left side of q_1q_2. Both C_1 and C_2 are incident to segment endpoints whose extension tree hits the left side of q_1q_2, and by Lemma 7 belong to L_2. Let p_1 and p_2 denote the segment endpoints of L_2 incident to C_1 and C_2, respectively.

Every segment endpoint incident to C_1 (resp, C_2) have the same color, otherwise L_1 and L_2 would be connected by a subgraph within C_1 (resp., C_2). We conclude that the graph G_1 has no edges within C_1 or C_2.

Consider the quadrilateral $\Delta = q_1q_2p_2p_1$. Let p' be the right-most segment endpoint in $\Delta \setminus \{q_1, q_2\}$. We argue that p' (and any segment endpoint in $\Delta \setminus \{q_1, q_2\}$) belongs to L_2: If $p' = p_1$ or $p' = p_2$, then obviously $p' \in L_2$. Otherwise p' lies to the right of both p_1 and p_2. Clearly, p' is a left segment endpoint and its extension (an x-monotone curve), must hit either q_1q_2 or the extension tree of p_1 or p_2. In any case, the root of the extension tree of p' lies on q_1q_2.

Finally, note that q_1 and q_2 have different colors, and so $p'q_1$ or $p'q_2$ is a bi-chromatic edge. It does not cross

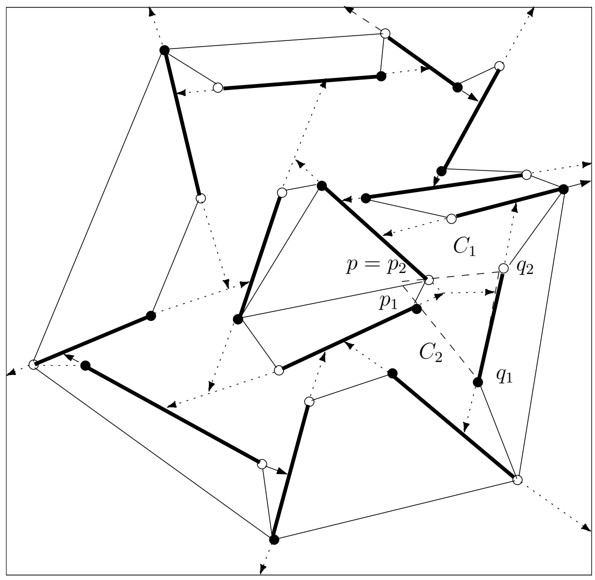

Figure 10: Connecting the components of G_1 in the second phase.

any edge of G_1, because the interior of Δ is disjoint from G_1. We connect L_1 and L_2 by augmenting G_1 with either $p'q_1$ or $p'q_2$.

If G_1 have more than one component, we iterate this step for the two leftmost compontents. Since every segment endpint in $\Delta \setminus \{q_1, q_2\}$ are in the same component of G_1, the quadrilaterals Δ arising from distinct steps are disjoint, and so we augmente G_1 with non-crossing edges.

Computational complexity. We can compute our color-conforming planar straight line spanning tree in $O(n \log n)$ time. We sort the right (resp., left) endpoints of the segments in $O(n \log n)$ time. Each sweepline algorithm is completed in $O(n \log n)$ time. The size of the resulting cell complex (together with orientation of segment extensions) is $O(n)$. We can add edges in all bi-colored cells in $O(n)$ total time. We can detect connected components of G_1 find complete the second phase of the algorithm in $O(n)$ time.

5 Open problems

We have shown that a color conforming spanning tree of a set of bi-chromatic line segments is always obtainable. What about a color conforming spanning tree with the minimum weight where the weight is computed as the sum of the Euclidean distances of the added edges? Given a set of points in the plane it is well known that a greedy algorithm always provides an optimal solution and the solution has no crossings. Bose and Toussaint showed that the minimum spanning tree that augments a set of line segments does not have any crossings [4]. However the minimum spanning tree of bi-chromatic line segments may introduce crossings, as is illustrated by the small example in Fig. 11. It would be interesting to explore methods for determining a color conforming minimum weight spanning tree of a set of bi-chromatic edges.

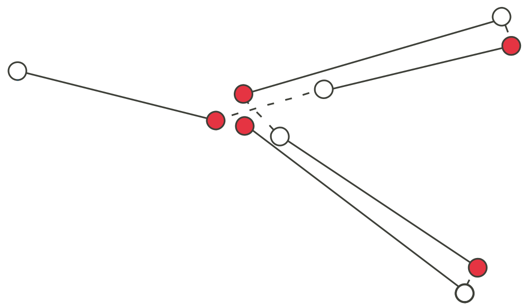

Figure 11: A color conforming minimum spanning tree for this example is not planar.

6 Acknowledgement

We would like to thank Henl Meijer for pointing out a flaw in our original proof of Lemma 3.

References

[1] J. Akiyama and J. Urrutia, Simple alternating path problem, *Discrete Math.* **84** (1) (1990), 101–103.

[2] M.Bern, Triangulations and mesh generation, In Handbook of Discrete and Computational Geometry 2nd Edition, Goodman J.E. and ORourke J (Eds), CRC Press, (2004), 563–582.

[3] P. Bose, M. E. Houle, and G.T. Toussaint, Every set of disjoint line segments admits a binary tree, *Discrete Comput Geom.* **26** (2001), 387–410.

[4] P. Bose and G. T. Toussaint, Growing a tree from its branches, *J. Algorithms* **19** (1995), 86–103.

[5] M. Hoffmann and Cs. D. Tóth, Alternating paths through disjoint line segments, *Inform. Proc. Letts.* **87** (2003), 287–294.

[6] M. Hoffmann and Cs. D. Tóth, Segment endpoint visibility graphs are Hamiltonian, *Comput. Geom. Theory Appl.* **26** (2003), 47–68.

[7] A. Kaneko, On the maximum degree of bipartite embeddings of trees in the plane, in *Discrete and Computational Geometry, JCDCG'98*, vol. 1763 of LNCS, Springer, 2098, 2000, pp. 166–171.

[8] A. Kaneko and M. Kano, Discrete geometry on red and blue points in the plane—a survey, in *Discrete and Computational Geometry, The Goodman-Pollack Festschrift*, vol. 25 of Algorithms and Combinatorics, Springer, 2003, 551–570.

[9] A. Kaneko and M. Kano, On paths in a complete bipartite geometric graph, in *Discrete and Computational Geometry, JCDCG'00*, vol. 2098 of LNCS, Springer, 2001, pp 187–191.

[10] A. Kaneko, M. Kano, and K. Yoshimoto, Alternating Hamiltonian cycles with minimum number of crossings in the plane, *Internat. J. Comput. Geom. Appl.* **10** (2000), 73–78.

[11] C. Larson, *Problem solving through problems*, Springer, 1983, pp. 200–201.

[12] J. Pach and E. Rivera-Campo, On circumscribing polygons for line segments, *Comput. Geom. Theory Appl.* **10** (1998), 121–124.